PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Rui Lang, et al	Technology Center: 2100
Serial No.: 10/646,851 Confirm: 1500	Group Art Unit: 2154
Filed: Aug. 23, 2003	Examiner: Michael E. Keefer
	Atty. Dkt. No.: 10830.0097NP
For: Multi-Protocol Sharable Virtual Storage Objects Management and Control	

APPEAL BRIEF TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450

Sir:

This Appeal Brief is in support of the Notice of Appeal filed March 7, 2008. A Petition for a One-Month Extension of Time to extend the time for the filing of this Appeal Brief to June 7, 2008 has been filed on even date with the filing of this Appeal Brief. Please deduct any deficiency in the required fees from EMC Corporation Deposit Account No. 05-0889.

I. REAL PARTY IN INTEREST

The real party in interest is EMC Corporation, by virtue of an assignment recorded at Reel 014480 Frame 0114.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF THE CLAIMS

Claims 1-49 have been presented for examination.

Claims 1-6, 12-18, 24-26, and 37-39 have been cancelled.

Claims 7-11, 19-23, 27-36, and 40-49 have been finally rejected, and are being appealed.

IV. STATUS OF AMENDMENTS

No amendment has been filed after the final Official Action of 12/10/2007.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The invention of appellants' independent claim 8 is a method of access to a storage object (65 in FIG. 2; specification, page 8 line 21 to page 9, line 11) in a first file server (21 in FIG. 1; page 6 line 18 to page 7 line 3) of a data processing network (20 in FIG. 1; page 6 lines 18-20). (Specification, page 3, lines 10-11.) The data processing network includes a client (23 in FIG. 1; page 6, lines 18-20), the first file server, and a second file server (22 in FIG. 1; page 6 line 18 to page 7 line 3). (Specification, page 3, lines 11-12.) The method includes the client using a block level access protocol ("iSCSI INITIATOR" 55 in FIG. 2; page 8 line 12 to page 9 line 7) over the network to access the storage object in the first file server; and the first file server accessing the storage object in the first file server by accessing a file (84 in FIG. 5; page 9 lines 4-11; page 10 lines 17-21) in the first file server containing data (87 in FIG. 5; page 10 line 22 to page 11 line 1) of the storage object. (Specification, page 3, lines 12-14.) The method also includes the first file server replicating (77 in FIG. 12; page 15 lines 13-18) a snapshot copy (76 in FIG. 12; page 15 lines 13-18) of the file from the first file server over the network to the second file server concurrent with the client using the block level access protocol over the network to write data to the storage object in the first file server. (Specification, page 4 line 23 to page 5 line 2; page 13 lines 10-23; page 15 lines 13-18.) The network is an IP network ("IP NETWORK" 20 in FIG. 1; page 6 lines 18-19). The client uses the block level access protocol over a first TCP/IP connection (112 in FIG. 12; page 15 lines 10-13) over the network to access the storage object in the first file server. (Specification, page 15 lines 10-13.) The client initiates the step of the first file server replicating the snapshot copy of the file over the network to the second file server by sending a command ("SNAP" in FIG. 10; page 22 lines 21-24) over a second TCP/IP connection (113 in FIG. 12; page 15 lines 10-15) to the first file server. (Specification, page 14 lines 4-12, page 15 lines 10-15, page 22 lines 21-24.) The method further includes the client pausing ("PAUSE" in FIG. 10, "SUSPEND" in step 123 of FIG. 13; page 22 lines 18-20, page 28 line 22 to page 29 line 3, page 29 lines 8-10) the step of writing of data to the storage object in the first file server after a commit operation ("SYNC" in step 123 in FIG. 13; page 28 line 22 to page 29 line 3, page 29 lines 8-10), and during the pause, the client performing (step 126 in FIG. 13; page 29 lines 4-13) the step of initiating the step of the first file server replicating the snapshot copy of the file from the first file server over the network to the second file server by sending the command over the second TCP/IP connection. (Specification, page 14 lines 13-18, page 15 lines 10-18.)

The invention of appellants' claim 8 permits the client to manage backup and replication of the storage object by using the snapshot copy and replication facilities during concurrent access to the storage object using the block level access protocol. (Specification, page 13, lines 18-20.) The invention of appellants' claim 8 enables the client to select a consistent view of the storage object to be replicated concurrent with write access to the storage object using the block level access protocol. (Specification, page 14, lines 4-12.)

The invention of appellants' independent claim 20 is a method of access to a virtual direct access storage device (65 in FIG. 2; specification, page 8 line 21 to page 9, line 3) in a first file server (21 in FIG. 1; page 6 line 18 to page 7 line 3) of a data processing network (20 in FIG.

1; page 6 lines 18-20). The data processing network includes a client (23 in FIG. 1; page 6, lines 18-20), the first file server, and a second file server (22 in FIG. 1; page 6 line 18 to page 7 line 3). (Specification, page 3, lines 11-12.) (Specification, page 3, lines 10-11.) Attributes (86 in FIG. 5; page 10 line 22 to page 12 line 13) and data (87 in FIG. 5; page 10 line 22 to page 11 line 1) of the virtual direct access storage device are stored in at least one file (84 in FIG. 5, 84 in FIG. 12; page 9 lines 4-10, page 10 lines 17-21) in the first file server. The method includes the client using a block level access protocol ("iSCSI INITIATOR" 55 in FIG. 2; page 8 line 10 to page 9 line 7) over the network to access the virtual direct access storage device in the first file server, and the first file server responding to commands in accordance with the block level access protocol ("SCSI TERMINATION" 64 in FIG. 2; 82 in FIG. 3; page 8 line 19 to page 9 line 11; page 9 lines 17-18) for access to the virtual direct access storage device in the first file server by accessing the attributes (steps 92 and 97 in FIG. 6; page 12 lines 16-18, page 13 lines 4-7) and data (step 95 in FIG. 6; page 12 line 21 to page 13 line 2) of the virtual direct access storage device in the first file server. (Specification, page 3 lines 18-23.) The method also includes the first file server providing access over the network to the virtual block storage device in the first file server in accordance with a file access protocol ("NFS" 41 in FIG. 2, "CIFS" 42 in FIG. 2; page 7 lines 4-9, page 9 lines 4-9) by accessing the at least one file in the first file server. (Page 3 line 23 to page 4 line 2.) The method also includes the first file server replicating (77 in FIG. 12; page 15 lines 13-18) a snapshot copy (76 in FIG. 12; page 15 lines 13-18) of the file from the first file server over the network to the second file server concurrent with the client using the block level access protocol over the network to write new data to the virtual direct access storage device in the first file server. (Specification, page 4 line 23 to page 5 line 2; page 13 lines 10-23; page 15 lines 13-18.) The network is an IP network ("IP NETWORK" 20 in FIG. 1; page 6 lines 18-20). The client uses the block level access protocol over a first TCP/IP connection (112 in FIG. 12; page 15 lines 10-13) over the network to access the virtual direct access storage device in the first file server. (Specification, page 15 lines 10-13.) The client initiates the step of the first file server replicating the snapshot copy of the at least one file by sending a command ("SNAP" in FIG. 10; page 22 lines 21-24) over a second TCP/IP connection (113 in FIG. 12; page 15 lines 10-15) to the first file server. (Specification, page 14 lines 4-12, page 15 lines 10-15, page 22 lines 21-24.) The method further includes the client pausing ("PAUSE" in FIG. 10, "SUSPEND" in step 123 of FIG. 13; page 22 lines 18-20, page 28 line 22 to page 29 line 3, page 29 lines 8-10) the writing of the new data to the virtual direct access storage device in the first file server after a commit operation ("SYNC" in step 123 in FIG. 13; page 28 line 22 to page 29 line 3, page 29 lines 8-10), and during the pause, the client performs (step 126 in FIG. 13; page 29 lines 4-13) the step of initiating the step of the first file server replicating the snapshot copy of the at least one file by sending the command over the second TCP/IP connection. (Specification, page 14 lines 13-18, page 15 lines 10-18.)

The invention of appellants' independent claim 27 is a network file server (21 in FIG. 1; page 6 line 18 to page 7 line 3). The network file server includes data storage (29 in FIG. 1; page 6 lines 20-22, page 7 lines 14-15), an interface for coupling the data storage to a data network (61 in FIG. 2; page 8 lines 17-18), and at least one processor (26 in FIG. 1, 26 in FIG. 2; page 6 lines 20-22) programmed for permitting clients in the data network to access the data storage in accordance with a plurality of access protocols ("NFS" 41 and "CIFS" 42 and "iSCSI TARGET" 63 in FIG. 2; page 7 lines 4-9, page 8 lines 19-21). (Specification, page 4, lines 4-7.) The data storage contains at least one file (84 in FIG. 5, 84 in FIG. 12; page 9 lines 4-11, page 10 lines 17-21) for storing file attributes (85 in FIG. 5; page 10 lines 17-21) and metadata (86 in FIG. 5; page 10 line 22 to page 12 line 13) defining a virtual direct access storage device (65 in FIG. 2; page 8 line 21 to page 9, line 3) and for storing data (87 in FIG. 5; page 10 line 22 to page 11 line 1) of the virtual direct access storage device. (Specification, page 4 lines 7-9.) The access protocols include at least one block level access protocol ("iSCSI TARGET" 63 in FIG. 2; page 8 line 18 to page 9 line 3) for access to the virtual direct access storage device by accessing the metadata (steps 92 and 97 of FIG. 6; page 12 lines 16-18, page 13 lines 4-7) and data (step 95 of FIG. 6; page 12 line 21 to page 13 line 2) of the virtual direct access storage device. The access protocols also include at least one file access protocol ("NFS" 41 or "CIFS" 42 in FIG. 2; page 7 lines 4-9, page 9 lines 4-9) for accessing the at least one file. (Specification, page 4 lines 11-12.) The metadata includes attributes of the virtual direct access storage device ("STORAGE OBJECT ATTRIBUTES" 86 in FIG. 5; page 10 line 22 to page 11 line 18). The attributes of the virtual direct access storage device and the data of the virtual direct access storage device are stored together in a single file (84 in FIG. 5, 84 in FIG. 12; page 9 lines 4-10, page 10 lines 17-21) in a file system ("UxFS" 44 in FIG. 12, "FILE SYSTEM" 88 in FIG. 12 as amended; page 7 lines 9-12, page 16 lines 17-20, page 25 lines 5-8). The attributes of the virtual direct access storage device include a specification of an internal organization of the virtual direct access storage device ("INTERNAL ORGANIZATION" of 86 in FIG. 5; page 11 line 13 to page 12 line 13) for mapping of the data of the virtual direct access storage device from the single file to the data storage, and the specification of the internal organization of the virtual direct access storage device is stored in the single file (page 10 line 22 to page 11 line 18).

The invention of appellants' independent claim 40 is a network file server (21 in FIG. 1; page 6 line 18 to page 7 line 3). The network file server includes data storage (29 in FIG. 1; page 6 lines 20-22, page 7 lines 14-15), an interface for coupling the data storage to an IP data network (61 in FIG. 2, 62 in FIG. 2; page 8 lines 17-19), and at least one processor (26 in FIG. 1, 26 in FIG. 2; page 6 lines 20-22) programmed for permitting clients in the data network to access the data storage in accordance with a plurality of access protocols ("NFS" 41 and "CIFS" 42 and "iSCSI TARGET" 63 in FIG. 2; page 7 lines 4-9, page 8 lines 10-21). (Specification, page 4, lines 14-16.) The data storage contains at least one file (84 in FIG. 5, 84 in FIG. 12; page 9 lines 4-10, page 10 lines 17-21) for storing file attributes (85 in FIG. 5; page 10 lines 17-21) and metadata (86 in FIG. 5; page 10 line 22 to page 12 line 13) defining a virtual SCSI direct access storage device (65 in FIG. 2; page 8 line 21 to page 9, line 3) and for storing data (87 in FIG. 5; page 10 line 22 to page 11 line 1) of the virtual direct access storage device. (Specification, page 4, lines 17-19.) The access protocols include a SCSI block level access protocol ("iSCSI TARGET" 63 in FIG. 2; page 8 line 18 to page 9 line 3) for client access to the virtual SCSI direct access storage device over the IP network by accessing the metadata (steps 92 and 97 of FIG. 6; page 12 lines 16-18, page 13 lines 4-7) and data (step 95 of FIG. 6; page 12 line 21 to page 13 line 2) of the virtual direct access storage device. (Specification, page 4 lines 19-21.) The access protocols further include at least one file access protocol ("NFS" 41 or "CIFS" 42 in

FIG. 2; page 7 lines 4-9, page 9 lines 4-9) for accessing the at least one file. (Specification, page 4 lines 21-23.) The network file server further includes a facility (77 in FIG. 12; page 15 lines 13-18) for remote replication of the at least one file over the IP network concurrent with client write access to the virtual SCSI direct access device over the IP network using the SCSI block level access protocol. (Specification, page 4 line 23 to page 5 line 2.) The metadata includes attributes ("STORAGE OBJECT ATTRIBUTES" 86 in FIG. 5; page 10 line 22 to page 11 line 18) of the virtual SCSI direct access storage device. The attributes of the virtual SCSI direct access storage device and the data of the virtual SCSI direct access storage device are stored together in a single file (84 in FIG. 5, 84 in FIG. 12; page 9 lines 4-10, page 10 lines 17-21) in a file system ("UxFS" 44 in FIG. 12, "FILE SYSTEM" 88 in FIG. 12 as amended; page 7 lines 9-12, page 16 lines 18-20, page 25 lines 5-8). The attributes of the virtual SCSI direct access storage device include a specification of an internal organization of the virtual SCSI direct access storage device ("INTERNAL ORGANIZATION" of 86 in FIG. 5; page 11 line 13 to page 12 line 13) for mapping of the data of the virtual SCSI direct access storage device from the single file to the data storage, and the specification of the internal organization of the virtual SCSI direct access storage device is stored in the single file (page 10 line 22 to page 11 line 18).

Appellants respectfully submit that none of the appellants' claims contain any "means plus function" or "step plus function" as permitted by 35 U.S.C. 112, sixth paragraph.

Appellants' dependent claims 7 and 19 each further define that the first TCP/IP connection (112 in FIG. 12; page 15 line 11) is concurrent with the second TCP/IP connection (113 in FIG. 12; page 15 line 12). (Specification, page 15 lines 10-18.)

Appellants' dependent claims 28 and 41 further define that the specified internal organization of the virtual direct access storage device includes a RAID level ("RAID LEVEL" in "STORAGE OBJECT ATTRIBUTES" 86 in FIG. 5; page 11 line 13 to page 12 line 3).

Appellants' dependent claims 29 and 42 further define that the specified internal organization of the virtual direct access storage device includes a striping pattern ("STRIPING" in "STORAGE OBJECT ATTRIBUTES" 86 in FIG. 5; page 11 line 13 to page 12 line 3).

Appellant's dependent claim 31 further defines that the interface is an IP interface (61 in FIG. 2; 62 in FIG. 2; page 8 lines 17-19), and the network file server is programmed to permit said one of the clients to write the new data to the virtual direct access storage device using the block level access protocol over a first TCP/IP connection (112 in FIG. 12; page 15 line 11) over the network for the writing of the new data to the virtual direct access storage device, and the network file server is programmed to initiate the copying of the file containing the data of the virtual direct access storage device over the network upon receipt of a command ("SNAP" in FIG. 10; page 22 lines 21-24) from the client over a second TCP/IP connection (113 in FIG. 12; page 15 line 12) over the network. (Specification, page 15 lines 10-18.)

Appellants' dependent claim 33 further defines that the interface is an IP interface (61 in FIG. 2, 62 in FIG. 2; page 8 lines 17-19), and the network file server includes an IP replication facility (77 in FIG. 12; page 15 lines 13-15) for replicating files (84 in FIG. 12; page 16 lines 17-20) from the data storage over the network (page 15 lines 19-21).

Appellants' dependent claim 43 further defines that the network file server is programmed to permit said at least one of said clients to write new data to the virtual SCSI direct access storage device using the block level access protocol over a first TCP/IP connection (112 in FIG. 12; page 15 line 11) over the network, and the network file server is programmed to initiate remote replication of said at least one file upon receipt of a command ("SNAP" in FIG. 10; page 22 lines 21-24) from said at least one of said clients over a second TCP/IP connection (113 in FIG. 12; page 15 line 12). (Specification, page 15 lines 10-18.)

Appellants' dependent claim 49 further defines that the network file server includes a snapshot copy facility (76 in FIG. 12; page 16 lines 17-20) for creating snapshot copies of said at least one file, and wherein the snapshot copy facility is coupled to the facility for remote replication for transmission of data from the snapshot copies over the IP network concurrent with client write access to the virtual SCSI direct access device over the IP network using the block level access protocol (page 15 lines 10-21).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 7-11 and 19-23 are unpatentable under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

2. Whether claims 27, 30, and 34-36 are unpatentable under 35 U.S.C. 102(e) as being anticipated by Chen et al. U.S. Pat. 7,076,509 B1.

3. Whether claims 7-11 and 19-23 are unpatentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 B1 in view of Baweja et al. U.S. Pat. 6,564,229 B1 and further in view of Lefebvre U.S. Pat. App. Pub. 2002/0010665 A1.

4. Whether claims 28-29 are unpatentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 B1 in view of Busser U.S. Pat. App. Pub. 2002/0095616 A1 and further in view of Hashemi U.S. Pat. 6,934,804 B2.

5. Whether claim 33 is unpatentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 B1 in view of Chen et al. U.S. Pat. 7,010,553 B2.

6. Whether claims 40 and 45-48 are unpatentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 B1 in view of Chen et al. U.S. Pat. 7,010,553 B2.

7. Whether claims 41 and 42 are unpatentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 B1 in view of Chen et al. U.S. Pat. 7,010,553 B2 and further in view of Busser U.S. Pat. App. Pub. 2002/0095616 A1 and further in view of Hashemi U.S. Pat. 6,934,804 B2.

8. Whether claims 31-32, 43-44 and 49 are unpatentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 B1 in view of Chen et al. U.S. Pat. 7,010,553 B2 and further in view of Bolosky et al. U.S. Pat. App. Pub. 2001/0052021 A1.

VII. ARGUMENT

1. Claims 7-11 and 19-23 are patentable under 35 U.S.C. 112, second paragraph, and particularly point out and distinctly claim the subject matter which applicant regards as the invention.

"The test for indefiniteness is whether one skilled in the art would understand the bounds of the claim when read in light of the specification. . . . If the claims read in light of the specification reasonably apprise those skilled in the art of the scope of the invention, §112 demands no more. . . . The degree of precision necessary for adequate claims is a function of the nature of the subject matter." Miles Labs., Inc., v. Shandon Inc., 997 F.2d 870, 875, 27 U.S.P.Q.2d 1123, 1126 (Fed. Cir. 1993)(citations omitted).

Paragraph 3 on page 2 of the Final Official Action says:

> Claims 8 and 20 state that the replicating and writing take place concurrently but Claims 8 and 20 also state that the writing is paused after a commit operation, and then during that pause, the replicating is initiated by the client. These two limitations are contradictory, as if the operations are not taking place at the same time, then they cannot be concurrent; especially as a commit operation comes at the -end- of an attempt to write, after which writing is completed by the client.

Appellants respectfully disagree. The claims do not specify that the commit operation, or any writing prior to the commit operation, is performed concurrent with replicating. Instead, a person of ordinary skill would understand that the commit operation terminates or completes a prior write operation.

Nor do the claims require any data actually to have been written concurrently to the storage object prior to the pausing of the writing of data to the storage object. For example, in the appellants' preferred embodiment, the client's pausing of the step of writing data to the storage object after the commit operation and the client's initiating the replication during the pause insure that all of the writing to the storage object concurrent with the replicating is performed after the client has initiated the replicating. However, the appellants' claims should not be limited to their preferred embodiment.

The terms "concurrent" and "pausing" should be given their plain and ordinary meanings consistent with their usage in the appellants' specification. "Concurrent" is commonly understood to be less restrictive than "simultaneous" so that two operations are concurrent if they are overlapping rather than co-extensive in time. "Pausing" is commonly understood to be less restrictive than "interrupting" so that "pausing" means "to stop or inhibit temporarily." Thus, under the plain and ordinary meanings of the terms "concurrent" and "pausing," the claims do not require all of the writing to the storage object after the commit operation to be simultaneous with the concurrent replicating of a snapshot copy of the storage object.

One skilled in the art understands from the appellants' specification that the terms "concurrent" and "pausing" are being used consistently with their plain and ordinary meanings as discussed above. For example, according to appellants' specification, page 14 lines 4 to 12:

As shown in FIG. 7, the client is provided with an application program called a virtual block device manager 71 for managing backup and replication of the client's storage object 65 in the data mover 26. In order to backup or replicate a consistent view of the storage object 65, write access to the storage object by the SCSI device driver is synchronized to the backup or replication process. For example, write access of the storage object 65 is paused at the completion of a synchronous write, a commit operation for a series of asynchronous writes, or a commit of a current transaction consisting of a series of write operations. During the pause, a snapshot copy operation is initiated for the backup or replication process.

As further described in appellants' specification, page 15 lines 10 to 18:

When a commit event has occurred and further writing over the iSCSI/TCP connection (112 in FIG. 12) is inhibited, a network block services (NBS) driver 74 in the client establishes a parallel and concurrent TCP connection (113 in FIG. 12) to a network block services server 75 in the data mover (21 in FIGS. 11 and 12). NBS control commands cause a snapshot copy facility 76 or an IP replication facility 77 to initiate a snapshot copy or IP replication process upon the storage object 65. The snapshot copy or IP replication process may continue as a background process concurrent with subsequent write access on a priority basis when the SCSI termination 64 executes SCSI write commands from the client's SCSI driver 54.

2. Claims 27, 30, and 34-36 are patentable under 35 U.S.C. 102(e) and are not anticipated by Chen et al. U.S. Pat. 7,076,509 B1.

"For a prior art reference to anticipate in terms of 35 U.S.C. § 102, every element of the claimed invention must be identically shown in a single reference." Diversitech Corp. v.

Century Steps, Inc., 7 U.S.P.Q.2d 1315, 1317 (Fed. Cir. 1988), quoted in In re Bond, 910 F.2d 831,15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990) (vacating and remanding Board holding of anticipation; the elements must be arranged in the reference as in the claim under review, although this is not an ipsis verbis test).

Appellants respectfully submit that claims 27, 30, and 34-36 are not anticipated by Chen et al. U.S. Pat. 7,076,509 because Chen et al. fails to disclose that the attributes of the virtual direct access storage device, which are stored in a single file with the data of the virtual direct access device, "include a specification of an internal organization of the virtual direct access storage device for mapping of the data of the virtual direct access storage device from the single file to the data storage, ..."

For example, as shown in appellants' FIG. 5, the single file 84 includes attributes 86 of the virtual direct access storage device, and data 87 of the virtual direct access device. The attributes 86 include "INTERNAL ORGANIZATION (E.G., RAID LEVEL, STRIPING)". As described in appellants' specification, page 11 lines 13-18:

> Moreover, the storage object attributes may include configuration information, such as a location (bus, target and LUN) of the storage object, and an internal organization of the storage object, such as a level of redundancy in an array of disk drives (RAID level) and a striping scheme. The specified internal organization of the storage object could be used as a guide or specification for mapping of the data storage area 87 of the container file 87 to storage in the cached disk array (49 in FIG. 2).

Page 4 of the Final Official Action says that with respect to the disclosure of Chen et al. U.S. Pat. 7,076,509, "Fig. 5 of fully incorporated application 10/216453 (now US 7107385) shows that the metadata does give an internal organization of the VLUN, showing where particular data is actually stored on a physical device." Paragraph 11 on page 17 of the Final Official Action says: "In the Rajan reference (US7107385), it is clear that in order for the file system to work as disclosed, the metadata contained in Fig. 5 must include where on the physical disks a specified set of data has been written, because the process of choosing what physical disks are used in the vdisk is automated. (See Col. 10., lines 51-67 - Col 11 lines 1-17 of Rajan). Since the server itself is choosing what parts of what physical disk make up each virtual disk, it is inherent that this data would be stored in the metadata in the inodes."

Appellants respectfully disagree, and submit that there is neither an explicit nor an "inherent" disclosure in the Rajan reference (US7107385) that the metadata contained in Fig. 5 includes where on the physical disks a specified set of data has been written.

In order to be inherent, a nondisclosed element must be inevitable. In other words, it is not sufficient if the element is sometimes present and sometimes absent; it must be inevitably present. See, for example, Tyler Refrigeration v. Kysor Indus. Corp., 777 F.2d 687, 689, 227 U.S.P.Q. 845, 846-47 (Fed. Cir. 1985)(it is inherent that a claimed embodiment of a reference patent maintains an air curtain during a defrost cycle). A retrospective view of inherency is not a substitute for some teaching or suggestion which supports the selection and use of the various elements in the particular claimed combination. In re Rijckaert, 9 F.3d 1531, 1534, 28 U.S.P.Q.2d 1955-1957 (Fed. Cir.

1993)(optimal condition of matching signal time exactly to recording time is not "inherent" in the prior art).

Chen does not disclose that the LUN file should include a specification of an internal organization for the virtual LUN such as a RAID level or a striping scheme or related information for mapping of the data of the virtual direct access storage device from the single file to the data storage. Instead, Chen Col. 10 lines 30-42 disclose that each volume 550 is constructed from an array of physical disks 530 organized as a RAID group. Chen Col. 10 lines 43-47 teaches that within each volume may be stored one or more virtual disks (vdisks). A vdisk is a special file type in a volume that derives from a plain (regular) file, but that has associated export controls and operation restrictions that support emulation of a disk. Thus, in Chen, the RAID redundancy or striping is disclosed as a property of the volume upon which the file system is built, rather than an attribute of the vdisk file.

Rajan US 7,107,385 says: “The file system provides a virtualization system that aggregates physical storage of a set of disks or portions (e.g., extents) of disks into a pool of blocks that can be dynamically allocated to form a vdisk. The file system also provides reliability guarantees for the vdisks in accordance with its underlying architecture. That is, the file system organizes its storage within volumes created among the managed disks. The vdisk is thereafter created as a storage object within a volume, and thus, inherits the underlying reliability configuration associated with that volume. The portions are aggregated and allocated as a vdisk with reliability guarantees in response to a request to create the vdisk from a user of the storage appliance and without further user involvement.” (Rajan, Abstract.) Thus, in Rajan, the RAID redundancy or striping also is disclosed as a property of the volume upon which the file system is built, rather than an attribute of the vdisk file.

Rajan Fig. 5 does not disclose an internal organization of the vdisk, such as a level of redundancy in an array of disk drives (RAID level) and a striping scheme, or any other vdisk attribute that would define where particular data is actually stored on a physical device. The attributes inode 540 in Rajan FIG. 5 includes numerous vdisk attributes of interest to a user or client as described in Rajan col. 13 line 31 to col. 14 line 3. The attributes inode 540 in Rajan FIG. 5 does not include an internal organization of the vdisk, such as a level of redundancy in an array of disk drives (RAID level) or a striping scheme, or any other vdisk attribute that would define where particular data is actually stored on a physical device.

There is no need in Rajan for the attributes inode 540 in Rajan FIG. 5 to include any attribute that would define where particular data is actually stored on a physical device because where particular data is actually stored on a physical device is a property of the volume upon which the file system is built. Since the virtualization system or a system administrator has chosen what parts of what physical disk make up the underlying volume (e.g., in FIG. 1, physical disks 130 are grouped to form VOL 1 and VOL 2; see col. 2 lines 10-43, and col. 6 lines 33-57), there is no need for this information to be stored as an attribute of the vdisk. Instead, this information could be stored in the disk storage layer 240 in Rajan FIG. 3, as described in col. 8 lines 7-9, or in volume manager functions of the file system layer 320 of FIG. 3, as described in col. 8, line 59 to col. 9 line 4. In any case, a vdisk is created within a volume after the file system 320 has organized it storage within volumes created among the managed disks. (Rajan, col. 10, lines 60-64.) The system administrator does not have to deal with reliability issues because a vdisk created on a volume "inherits" the reliability guarantees of the file system and its underlying volume. (Rajan, col. 11, lines 18-35.) For example, if a vdisk is created on an underlying volume configured from disks that are organized as RAID 4, the operating system need not access any metadata stored in the vdisk file to determine an internal RAID organization to resize the vdisk because the operating system simply chooses logical blocks from the underlying volume and these logical blocks will be RAID 4 and not RAID 6 logical blocks because the underlying volume consists of RAID 4 logical blocks and not RAID 6 logical blocks. (Cf. final Official Action paragraph 11 on page 18.)

The fact that in Chen/Rajan the virtualization system 300 creates a vdisk in such a manner that is transparent to the user (Rajan '385 col. 11 lines 7-18) teaches away from including, in a container file together with data of the virtual direct access device, the RAID redundancy, striping, and related information. Rajan says: "This 'inherited' reliability approach of the multi-protocol appliance simplifies management of the vdisk because a user (system administrator) does not have to address the reliability issue on a storage object (vdisk) basis. Rather, the system administrator need merely render global choices of reliability with respect to an entire volume." (Rajan, col. 3, lines 43-49.)

In short, only the appellants' novel disclosure teaches that the RAID redundancy, striping, and related information should be stored in the container file for a virtual direct access storage device, for example, so that this information is transported with the other contents of the container file to a remote file server when the container file is replicated to the remote file server.

3. Claims 7-11 and 19-23 are patentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 B1 in view of Baweja et al. U.S. Pat. 6,564,229 B1 and further in view of Lefebvre U.S. Pat. App. Pub. 2002/0010665 A1.

Claims 8-11 and 20-23

The policy of the Patent and Trademark Office has been to follow in each and every case the standard of patentability enunciated by the Supreme Court in Graham v. John Deere Co., 148 U.S.P.Q. 459 (1966). M.P.E.P. § 2141. As stated by the Supreme Court:

> Under § 103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or nonobviousness of the subject matter is determined. Such secondary considerations as commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented. As indicia of obviousness or nonobviousness, these inquiries may have relevancy.

148 U.S.P.Q. at 467.

The problem that the inventor is trying to solve must be considered in determining whether or not the invention would have been obvious. The invention as a whole embraces the structure, properties and problems it solves. In re Wright, 848 F.2d 1216, 1219, 6 U.S.P.Q.2d 1959, 1961 (Fed. Cir. 1988).

The appellants' independent claims 8 and 20 define a method of replication of a snapshot copy of a storage object container file from a first file server over an IP data network to a second file server concurrent with a client using a block level access protocol over the IP network for writing data to the storage object in the first file server. The client uses the block level access protocol over a first TCP/IP connection over the network to access the storage object in the first file server, and the client initiates the step of the first file server replicating the snapshot copy of the file over the network to the second file server by sending a command over a second TCP/IP connection to the first file server. Moreover, the client pauses the step of writing of data to the storage object in the first file server after a commit operation, and during the pause, the client performs the step of initiating the step of the first file server replicating the snapshot copy of the file from the first file server over the network to the second file server by sending the command over the second TCP/IP connection.

The advantage of the invention of appellants' independent claims 8 and 20 is set out in appellants' specification on page 13 line 18 to page 14 line 12 et seq.:

> In the data processing system of FIG. 2, it is desired to permit the client 23 to manage backup and replication of its SCSI storage object in the data mover 26 during concurrent access to the storage object using the SCSI over IP protocol. For example, while the client 23 writes data to the data mover 26, the data mover 26 replicates the data to the second network file server 22 in FIG. 1 by transmitting a copy of the data over the IP network 20 using the NFS or CIFS protocols. One way of doing this is to provide a parallel and concurrent TCP connection between the client 23 and the data mover 26 for control of snapshot

> copy and IP replication applications in the data mover 26. This method is described below with reference to FIGS. 7 to 14.
>
> As shown in FIG. 7, the client is provided with an application program called a virtual block device manager 71 for managing backup and replication of the client's storage object 65 in the data mover 26. In order to backup or replicate a consistent view of the storage object 65, write access to the storage object by the SCSI device driver is synchronized to the backup or replication process. For example, write access of the storage object 65 is paused at the completion of a synchronous write, a commit operation for a series of asynchronous writes, or a commit of a current transaction consisting of a series of write operations. During the pause, a snapshot copy operation is initiated for the backup or replication process.

Chen U.S. Pat. 7,076,509 discloses a system and method for restoring a vdisk from a snapshot without the need to copy every individual block or inode from the snapshot. (Chen, Abstract.) A vdisk object is a special file type that is implemented in a multi-protocol storage appliance by a virtualization system and translated into an emulated disk as viewed by the SAN clients. (Chen, col. 8, lines 30-33.) Client systems generally utilize file based access protocols (CIFS or NFS over TCP/IP) when accessing information (in the form of files or directories) over a NAS based network. (Chen, col. 9, lines 6-21.) The multi-protocol storage appliance supports various SCSI-based protocols used in SAN deployments, including SCSI encapsulated over TCP (iSCSI) and SCSI encapsulated over FC (FCP). (Chen, col. 9, lines 55-58.) Within each volume may be stored one or more virtual disks (vdisks). A vdisk is a special file type in a volume that derives from a plain (regular) file, but that has associated export controls and operation restrictions that support emulated of a disk. In the illustrative embodiment, a vdisk is a multi-inode object that comprising a special file inode and a set of stream inodes that are managed as a single, encapsulated storage object within a file system of a storage system. (Chen, col. 10, lines 43-50.)

Regarding differences between Chen U.S. Pat. 7,076,509 and the appellants' independent claim 8, Chen does not explicitly disclose replication of a snapshot copy of the file from the first file server over the network to a second file server concurrent with the client using the block level access protocol over the network to write data to the storage object in the first file server. With respect to this difference, page 6 of the final Official Action cites Chen col. 13 lines 43-49 for disclosure of snapshot copy and restore of a vdisk, and Chen's incorporated reference Hitz et al. (U.S. Pat. 5,819,292) col. 1 lines 26-32 for disclosure of allowing the file server to remain on-line during the back-up of a clone or snapshot copy of the active file system.

Chen does not disclose the client initiating the step of the first file server replicating the snapshot copy of the file over the network to the second file server by sending a command over a second TCP/IP connection to the first file server. Nor does Chen disclose the client pausing the step of writing data to the storage object in the first file server after a commit operation, and during the pause, the client performing the step of initiating the step of the first file server replicating the snapshot copy of the file from the first server over the network to the second file server by sending the command over the second TCP/IP connection.

Page 9 of the final Official Action cites Baweja for a general concept of pausing data writing and during the pause initiating the copying of the file.

Baweja discloses a move/copy interface with a pause feature that allows the user to pause and subsequently resume a move or copy command. In one embodiment, the pause tool saves an index, source file name, target file name, block size and block number so that the operation can be subsequently resumed, After a pause has been requested, a "resume" command button appears on the user interface. The user selects the "resume" button to resume processing. Extended periods between a pause and subsequent resume are provided by saving the pause data to a data file. Another embodiment pauses a copy operation over a computer network, such as the internet, suspending the source computer's sending of blocks of data comprising the source file until the resume operation is requested. The user can repeatedly pause and resume the copy operation in order to free system resources in order to perform other operations. (See Baweja, Abstract.)

The Official Action cites Lefebvre for a general concept of a client requesting a replication of snapshot data over a TCP connection.

FIG. 1 of Lefebvre discloses a real time global tariff and import data system 120 connected via the Internet 104 to client devices 102. (Lefebvre paragraph [0031].) The data system 120 includes application servers 132, 134 and data layer servers 142, 144 accessing a database 146 of the tariff and import data stored in a set of shard RAID external disks. (Lefebvre paragraphs [0039] and [0043].) In the preferred form, the data layer servers 142 and 144 of FIG. 1 are Microsoft SQL servers, clustered using standard clustering technology (e.g., such as that provided by Microsoft Corporation of Redmond Wash.). (Lefebvre paragraph [0043].) SQL servers allow users to replicate data from one SQL Server to another SQL server, and to several other types of databases by different makers (e.g., Oracle, Sybase, or IBM DB2). The SQL Server replication function is based on the "publish and subscribe" model in which one database information server plays the role of a "publisher" while the others play the role of "subscribers." (Lefebvre, paragraph [0081].) Snapshot replication takes a snapshot of data to be published at a given moment in time. These snapshots can be taken according to a plan or upon request. (Lefebvre, paragraph [0086].)

With respect to Lefebvre, page 10 of the final Official Action says: "It is inherent that this action [of a client requesting replication of snapshot data over a TCP connection] would take place over a different TCP socket than that used by iSCSI as they are different protocols which would use different TCP sockets to perform their tasks." Page 10 of the Official Action concludes: "It would have been obvious to one of ordinary skill in the art at the time of the invention to [combine] Chen and Baweja with the general concept of a client requesting a replication of snapshot data over a TCP connection as taught by Lefebvre in order to back the data snapshots more securely backed up of-site."

Appellants respectfully disagree with the conclusion in the Official Action because neither Chen nor Baweja nor Lefebvre individually nor in combination suggest that a client using block access to write data to a storage object is so different from the client requesting replication of a snapshot copy of the storage object that one of ordinary skill would recognize that the client should do the writing over a first TCP/IP connection to the first file sever and the client should do the requesting of the replication over a second TCP connection. In particular, it is not "inherent" for the client to request replication using a different protocol or a different TCP/IP connection because the block access protocol for accessing and writing to the storage object could be extended to include a command to initiate replication of a snapshot of the storage object. For example, a client command for a synchronous write or a "commit" operation could include a flag indicating whether or not a snapshot for replication should be taken upon the committed data. As disclosed in the appellants' specification, page 14 lines 6 to 12, the appellants have found that it is more convenient for the client to pause the writing of data to the storage object for the purpose of synchronization in order to backup or replicate a consistent view of the storage object, because this can be done using existing services of Microsoft Exchange or a UNIX API. Thus, it is a matter of convenient software interoperability rather than "inherency" or necessity.

Appellants also respectfully point out that "an inherent feature of a reference may be relied upon to establish obviousness only if such inherency would have been obvious to one of ordinary skill in the art." Kloster Speedsteel AB v. Crucible, Inc., 793 F.2d 1565, 1576,230 U.S.P.Q. 81, 88 (Fed. Cir. 1986), cert. denied, 479 U.S. 1034 (1987). There is no disclosure in Lefebvre of an iSCSI connection between a client and a server, and instead communication between clients and servers appears to use HTTP or XML and SQL. (See Lefebvre XML strings 702 and 704 in FIG. 7A, steps 718 to 734 in FIG. 7B; Lefebvre paragraphs [0059] and [0113].) There is no disclosure in Lefebvre that a subscriber server would use more than one protocol or socket for requesting and obtaining a snapshot from a publisher server. (See Lefebvre, paragraphs [0083] and [0086].) Thus, the inherency referred to on page 10 of the final Official Action apparently is referring to inherency in a hypothetical combination of features from multiple references. Such inherency cannot be relied upon as a motivation or suggestion for picking and choosing the particular features from the references in the first instance because such inherency would not have existed and thus would not have been obvious to one of ordinary skill prior to making the hypothetical combination.

Appellants respectfully submit that the cited art as a whole does not suggest the specific combination defined in appellants' claim 8. Baweja discloses a move/copy interface with a pause feature that allows the user to pause and subsequently resume a move or copy command in order to free system resources for performing other operations during the pause. In appellants' claim 8, a client pauses a write operation in a server to initiate a copy operation, instead of pausing a copy operation to perform a write operation. (Moreover, as disclosed in appellants' specification, page 14, lines 6 to 12, the appellants' write operation is being paused for synchronization rather than for freeing system resources.) Lefebvre discloses a client requesting replication of snapshot data over a TCP connection but does not suggest that the request would occur using a different protocol or socket than the transfer of data. In contrast, the appellants' two TCP connections are for access at different levels upon the data - block level access or file level access. In addition, appellants' claims 8 and 20 also recite functions at the client in addition to the access at different levels in the first file server. Activity on the client is being coordinated with the concurrent block level write and file snapshot copy and replication process on the server using the two separate TCP/IP connections. Therefore, a person of ordinary skill would not have been motivated by Baweja and Lefebvre to modify Chen to further include replicating a snapshot copy of the file from the first file server over the network to a second file server concurrent with the client using the block level access protocol over the network to write data to the storage object in the first file server, the client initiating the step of the first file server replicating the snapshot copy of the file over the network to the second file server by sending a command over a second TCP/IP connection to the first file server, the client pausing the step of writing data to the storage object in the first file server after a commit operation, and during the pause, the client performing the step of initiating the step of the first file server replicating the snapshot copy of the file from the first server over the network to the second file server by sending the command over the second TCP/IP connection.

"[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." In re Kahn, 441 F. 3d 977, 988 (Fed. Cir. 2006). A fact finder should be aware of the distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning. See KSR v. Teleflex, 550 U.S. __ (2007), citing Graham, 383 U. S. at 36 (warning against a "temptation to read into the prior art the teachings of the invention in issue" and instructing courts to "guard against slipping into the use of hindsight.").

Claims 7 and 19

Appellants' dependent claims 7 and 19 are dependent upon claims 8 and 20, and therefore they are patentable over Chen, Baweja, and Lefebvre for the reasons given above with respect to claims 8 and 20. In addition, claims 8 and 20 further define that the first TCP/IP connection is concurrent with the second TCP/IP connection. This further distinguishes

Lefebvre (paragraphs [0083] and [0086]). If a subscriber server were to use a first TCP connection for requesting a snapshot from a publisher server and a second TCP connection for receiving snapshot data from the publisher server, there would be no need for the two TCP connections to be concurrent.

4. Claims 28-29 are patentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 B1 in view of Busser U.S. Pat. App. Pub. 2002/0095616 A1 and further in view of Hashemi U.S. Pat. 6,934,804 B2.

Claims 28 and 29 are dependent upon claim 27. Claim 28 further defines that the specified internal organization of the virtual direct access storage device includes a RAID level. Claim 29 further defines that the specified internal organization of the virtual direct access storage device includes a striping pattern. This further distinguishes Chen et al. because neither Chen et al. (e.g. FIG. 7) nor its incorporated Rajan et al. U.S. 7,107,385 discloses that a RAID level or striping pattern is included in the vdisk attributes in the vdisk file (i.e., the LUN INODE 710 and its descendant inodes in Chen FIG. 7 or the LUN INODE 510 and its descendant inodes in Rajan FIG. 5). As discussed above with reference to claim 27, Chen et al. and Rajan et al. teach away from including a RAID level in the vdisk attributes, because Chen and Rajan disclose that the RAID redundancy or striping is a property of the volume upon which the file system is built, rather than an attribute of the vdisk file. Rajan in particular teaches that the file system provides reliability guarantees for the vdisks in accordance with its underlying architecture. That is, the file system organizes its storage within volumes created among the managed disks. The vdisk is thereafter created as a storage object within a volume, and thus, inherits the underlying reliability configuration associated with that volume. The portions are aggregated and allocated as a vdisk with reliability guarantees in response to a request to create the vdisk from a user of the storage appliance and without further user involvement. (Rajan, Abstract.)

Page 10 of the final Official Action cites Busser for the general concept of storing RAID level information in metadata files.

Busser US 2002/0095616 in FIG. 2 shows stored data 90 in a disk of a RAID storage system. The stored data is divided into two categories, disk metadata 100, and customer data 200. The disk metadata 100 contains data that the controller 26 of the RAID storage system uses to assist RAID operation and management. (Busser, page 3, paragraph [0030].) The disk metadata 100 includes the RAID level 132 of the array. The RAID level 132 is the level that correspondents to the particular RAID architecture that is in use on a particular array. (Busser, page 3, paragraph [0032].) The customer data 200 is data stored on the RAID system 10 which was sent by the host computer 30. This customer data 200 may include a variety of information from the host computer 30, such as software programs, personal data, or financial data, to name a few. (Busser, page 3, paragraph [0029].)

The RAID level information in Busser paragraph [0029] is on a disk header (FIG. 2) with partition information and is not stored in a metadata file. Apparently the controller of the RAID system 10 accesses the disk metadata 100 directly, without use of a file system, and the customer data 200 may include files accessible to the host. (See Busser, page 4, paragraph

[0041]; "the host and controller determines the file locations on the drives within the array.") Thus, Busser does not disclose that the RAID level 132 is in any file such as any of the files accessible to the host. Consequently, a proper combination of Busser with Chen/Rajan would not result in the invention defined in appellants' claims 27 and 28. There is nothing in Busser that suggests modification of the teaching in Chen et al. that a logical volume is configured for a RAID level and then a file system is built on the logical volume so that the RAID level is not a property of a file or a virtual LUN.

Hashemi (US 6,934,804) discloses a general concept of striping data at the RAID level, but Hashemi does not suggest that striping information should be stored in a single file together with the attributes and the data of a virtual direct access storage device. For example, Hashemi col. 10 line 66 says:

> The distribution of data and spare storage in array A300 is performed beneath the RAID layout. Each of the LUNs in the array A300 can be further partitioned or concatenated with other LUNs to form smaller or larger LUNs for defining other RAID attributes like those of striping and/or redundancy, in accordance with another embodiment of the invention.

Hashemi does not disclose storing a parameter used for control of striping. Instead, the striping appears to be defined by how a large LUN would be built up by concatenation of partitions of disks in the RAID array. Therefore, neither Busser nor Hashemi suggest that Chen should be modified to arrive at the invention of appellants' claims 28 and 29.

5. Claim 33 is patentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553.

Appellants' claim 33 is dependent upon claim 27, and further defines that the interface is an IP interface and the network file server includes an IP replication facility for replicating files from the data storage over the network. Page 12 of the final Official Action cites Chen et al. U.S. 7,010,553 for the general concept of backing file up over a network. However, Chen et al. U.S. Pat. 7,010,553 does not disclose the elements in the base claim 27 that are missing from Chen et al. U.S. 7,076,509. In particular, Chen et al. U.S. Pat. 7,010,553 does not provide a motivation for modifying Chen et al. U.S. 7,076,509 to include the specification of the internal organization of the virtual direct access storage device in the single file that stores the attributes of the virtual direct access storage device together with the data of the virtual direct access storage device. Therefore, claim 33 is patentable over Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 for the reasons given above with respect to the base claim 27.

6. Claims 40 and 45-48 are patentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553.

Appellants' claims 45-48 are each dependent upon claim 40. Claim 40 is an independent claim defining a network file server. Claim 40 includes limitations similar to those found in appellants' claims 27 and 33, and therefore claims 40 and 45-48 are patentable over Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 for the reasons given above with respect to appellants' claims 27 and 33. Neither Chen et al. U.S. Pat. 7,076,509 nor Chen et al. U.S. Pat. 7,010,553 provides a motivation or suggestion for including a specification of an internal organization of the virtual SCSI direct access storage device in the single file that stores the attributes of the virtual SCSI direct access storage device together with the data of the virtual SCSI direct access storage device.

7. Claims 41 and 42 are patentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 B1 in view of Chen et al. U.S. Pat. 7,010,553 B2 and further in view of Busser U.S. Pat. App. Pub. 2002/0095616 A1 and further in view of Hashemi U.S. Pat. 6,934,804 B2.

Appellants' claims 41 and 42 are dependent upon claim 40 and are similar to claims 28 and 29. Claim 41 further defines that the specified internal organization of the virtual SCSI direct access storage device includes a RAID level. Claim 42 further defines that the specified internal organization of the virtual SCSI direct access storage device includes a striping pattern. Therefore, claims 41 and 42 are patentable over Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 and further in view of Busser and Hashemi for the reasons give above with respect to claims 28 and 29. Neither Chen et al. U.S. Pat. 7,076,509 (e.g. FIG. 7) nor its incorporated Rajan et al. U.S. 7,107,385 discloses that a RAID level or striping pattern is included in the vdisk attributes in the vdisk file (i.e., the LUN INODE 710 and its descendant inodes in Chen FIG. 7 or the LUN INODE 510 and its descendant inodes in Rajan FIG. 5). As discussed above with reference to claim 27, Chen et al. U.S. Pat. 7,076,509 and Rajan et al. teach away from including a RAID level in the vdisk attributes, because Chen and Rajan disclose that the RAID redundancy or striping is a property of the volume upon which the file system is built, rather than an attribute of the vdisk file. Busser discloses RAID level information on a disk header (FIG. 2) with partition information for use by a controller of a RAID system. Thus, Busser does not disclose that the RAID level 132 is in any file such as any of the files accessible to the host. Consequently, a proper combination of Busser with Chen/Rajan would not result in the invention defined in appellants' claims 41 and 42. There is nothing in Busser that suggests modification of the teaching in Chen et al. that a logical volume is configured for a RAID level and then a file system is built on the logical volume, so that the RAID level is not a property of a file or a virtual LUN. Hashemi (US 6,934,804) discloses a general concept of striping data at the RAID level, but Hashemi does not suggest that striping information should be stored in a single file together with the attributes and the data of a virtual direct access storage device. Therefore, neither Busser nor Hashemi suggest that Chen should be modified to arrive at the invention of appellants' claims 41 and 42.

8. Claims 31-32, 43-44 and 49 are patentable under 35 U.S.C. 103(a) over Chen et al. U.S. Pat. 7,076,509 B1 in view of Chen et al. U.S. Pat. 7,010,553 B2 and further in view of Bolosky et al. U.S. Pat. App. Pub. 2001/0052021 A1.

Claims 31-32

Claims 31 and 32 are dependent upon claim 27. Claim 32 is dependent upon claim 31. Claim 31 further distinguishes the proposed combination of Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 by defining that the network file server is programmed to permit the client to write the new data to the virtual direct access storage device using the block level access protocol over a first TCP/IP connection over the network for the writing of the new data to the virtual direct access storage device, and the network file server is programmed to initiate the copying of the file containing the data of the virtual direct access storage device over the network upon receipt of a command from the client over a second TCP/IP connection.

Bolosky et al. discloses a wire protocol providing message formats for creating multiple network connections between a media server and a client. These multiple network connections may include a control link connection for passing control information and a data funnel connection for passing data of multiple media. The data funnel connection may be a multipoint-to-point connection that connects multiple data servers with the client. The protocol facilitates multiple requests being concurrently outstanding and asynchronous processing of requests. The protocol is designed to exist on top of a transport protocol layer. (Bolosky, Abstract.) The wire protocol is utilized to create a control connection between the media server and the client to facilitate exchange of control information. The wire protocol is also used to create a data connection between the media server and the client that facilitates the exchange of data between the media and the client at a rate substantially equal to a rate at which the data is consumed by the client. (Bolosky, paragraph [0003].)

Bolosky fails to provide any motivation or suggestion for modifying the proposed combination of Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 to provide the elements of the independent base claim 27 that are missing from Chen et al. U.S. Pat. 7,076,509 and Chen et al. U.S. Pat. 7,010,553. In particular, there is nothing in Bolosky suggesting that the specification of an internal organization of the virtual direct access storage device should be stored in the single file together with the data of the virtual direct access device. Therefore, claims 31 and 32 are patentable over Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 because of the limitations incorporated by reference from the base claim 27.

Regarding claims 31 and 32, page 17 of the Official Action concludes: "It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Chen et al. (US7076509 B1) and Chen et al. (US 7010553 B2) with the general concept of a remote client using a command over a separate TCP/IP connection as taught by Bolosky in order to facilitate efficient and useful communications between clients and servers. (Bolosky, [0002] last sentence)." Appellants respectfully disagree, because the environment of a Bolosky's media server, in which a client sends multiple requests concurrently for data of multiple media from the server and the server sends the media data to the client at a rate substantially equal to the rate at which the client consumes the media data, is substantially different from the environment of appellants' server, in which a client sends block level access commands and snapshot/replicate commands to the server.

As discussed above, it is not necessary for a separate protocol or a separate TCP/IP connection to be used for a client to send snapshot/replicate commands to the server, and in the substantially different environment of appellants' data processing system, snapshot/replicate commands could be included in the block access protocol, for example, as a flag in a synchronous write or commit command. Thus, appellants respectfully submit that the cited references as a whole provide an insufficient motivation or suggestion for modifying the proposed combination of Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 to arrive at the invention of appellants' claims 31 and 32.

Claims 43-44

Claims 43-44 are dependent upon claim 40. Claim 43 is dependent upon claim 40. Claim 43 further distinguishes the combination of Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 by defining that the network file server is programmed to permit the client to write new data to the virtual SCSI direct access storage device using the block level access protocol over a first TCP/IP connection over the network, and the network file server is programmed to initiate remote replication o the file upon receipt of a command from the client over a second TCP/IP connection over the network.

Bolosky fails to provide any motivation or suggestion for modifying the proposed combination of Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 to provide the elements of the independent base claim 40 that are missing from Chen et al. U.S. Pat. 7,076,509 and Chen et al. U.S. Pat. 7,010,553. In particular, there is nothing in Bolosky to suggest that the specification of the internal organization of the virtual SCSI direct access storage device should be stored in the single file together with the data of the virtual direct access device. Therefore, claims 43-44 are patentable over Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 because of the limitations of claims 43-44 that are incorporated by reference from the base claim 40.

Regarding claims 43 and 44, page 17 of the Official Action concludes: "It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Chen et al. (US7076509 B1) and Chew et al. (US 7010553 B2) with the general concept of a remote client using a command over a separate TCP/IP connection as taught by Bolosky in order to facilitate efficient and useful communications between clients and servers. (Bolosky, [0002] last sentence)." Appellants respectfully disagree, because the environment of a Bolosky's media server, in which a client sends multiple requests concurrently for data of multiple media from the server and the server sends the media data to the client at a rate substantially equal to the rate at which the client consumes the media data, is substantially different from the environment of appellants' server, in which a client sends block level access commands and snapshot/replicate commands to the server.

As discussed above, it is not necessary for a separate protocol or a separate TCP/IP connection to be used for a client to send snapshot/replicate commands to the server, and in the substantially different environment of appellants' data processing system, snapshot/replicate commands could be included in the block access protocol, for example, as a flag in a synchronous write or commit command. Therefore, appellants respectfully submit that the cited references as a whole fail to provide a sufficient motivation or suggestion for modifying the proposed combination of Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 to arrive at the invention of appellants' claims 43 and 44.

Claim 49

Claim 49 is dependent upon independent claim 40. Bolosky fails to provide any motivation or suggestion for modifying the proposed combination of Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 to provide the elements of the independent base claim 40 that are missing from Chen et al. U.S. Pat. 7,076,509 and Chen et al. U.S. Pat. 7,010,553. In particular, there is nothing in Bolosky to suggest that the specification of the internal organization of the virtual SCSI direct access storage device should be stored in the single file together with the data of the virtual direct access device. Therefore, claim 49 is patentable over Chen et al. U.S. Pat. 7,076,509 in view of Chen et al. U.S. Pat. 7,010,553 because of the limitations of claim 49 that are incorporated by reference from the base claim 40.

In view of the above, the rejections of the claims should be reversed.

Respectfully submitted,

/ Richard C. Auchterlonie /

Richard C. Auchterlonie
Reg. No. 30,607

NOVAK DRUCE & QUIGG, LLP
1000 Louisiana, 53rd Floor
Houston, TX 77002
713-571-3400

VIII. CLAIMS APPENDIX

The claims involved in this appeal are as follows:

7. The method as claimed in claim 8, wherein the first TCP/IP connection is concurrent with the second TCP/IP connection.

8. In a data processing network including a client, a first file server, and a second file server, a method of access to a storage object in the first file server, said method comprising:

the client using a block level access protocol over the network to access the storage object in the first file server; and

the first file server accessing the storage object in the first file server by accessing a file in the first file server containing data of the storage object;

which includes the first file server replicating a snapshot copy of the file from the first file server over the network to the second file server concurrent with the client using the block level access protocol over the network to write data to the storage object in the first file server;

wherein the network is an IP network, the client uses the block level access protocol over a first TCP/IP connection over the network to access the storage object in the first file server, and the client initiates the step of the first file server replicating the snapshot copy of the file over the network to the second file server by sending a command over a second TCP/IP connection to the first file server; and which includes the client pausing the step of writing of data to the storage object in the first file server after a commit operation, and during the pause, the client performing the step of initiating the step of the first file server replicating the snapshot copy of the file from the first file server over the network to the second file server by sending the command over the second TCP/IP connection.

9. The method as claimed in claim 8, which includes the first file server also providing access to the storage object in the first file server over the network by means of a file access protocol over the network, the file access protocol accessing the file in the first file server containing the data of the storage object in the first file server.

10. The method as claimed in claim 9, wherein the file access protocol is the Network File System (NFS) protocol.

11. The method as claimed in claim 9, wherein the file access protocol is the Common Internet File System (CIFS) protocol.

19. The method as claimed in claim 20, wherein the first TCP/IP connection is concurrent with the second TCP/IP connection.

20. In a data processing network including a client, a first file server, and a second file server, a method of access to a virtual direct access storage device in the first file server, attributes and data of the virtual direct access storage device being stored in at least one file in the first file server, said method comprising:

the client using a block level access protocol over the network to access the virtual direct access storage device in the first file server, the first file server responding to commands in accordance with the block level access protocol for access to the virtual direct access storage device in the first file server by accessing the attributes and data of the virtual direct access storage device in the first file server; and

the first file server providing access over the network to the virtual block storage device in the first file server in accordance with a file access protocol by accessing said at least one file in the first file server;

which includes the first file server replicating a snapshot copy of said at least one file from the first file server over the network to the second file server concurrent with the client using the block level access protocol over the network to write new data to the virtual direct access storage device in the first file server;

wherein the network is an IP network, the client uses the block level access protocol over a first TCP/IP connection over the network to the first file server to access the virtual direct access storage device in the first file server, and the client initiates the step of the first file server replicating the snapshot copy of said at least one file by sending a command over a second TCP/IP connection to the first file server; and which includes the client pausing the writing of the new data to the virtual direct access storage device in the first file server after a commit operation, and during the pause, the client performs the step of initiating the step of the first file server replicating the snapshot copy of said at least one file by sending the command over the second TCP/IP connection to the first file server.

21. The method as claimed in claim 20, wherein the network is an IP network, and the block level access protocol is the Small Computer System Interface (SCSI) protocol.

22. The method as claimed in claim 20, wherein the file access protocol is the Network File System (NFS) protocol.

23. The method as claimed in claim 20, wherein the file access protocol is the Common Internet File System (CIFS) protocol.

27. A network file server comprising:

data storage;

an interface for coupling the data storage to a data network; and

at least one processor programmed for permitting clients in the data network to access the data storage in accordance with a plurality of access protocols;

the data storage containing at least one file for storing file attributes and for storing metadata defining a virtual direct access storage device and for storing data of the virtual direct access storage device;

the access protocols including at least one block level access protocol for access to the virtual direct access storage device by accessing the metadata and data of the virtual direct access storage device; and

the access protocols including at least one file access protocol for accessing said at least one file;

wherein the metadata includes attributes of the virtual direct access storage device, and the attributes of the virtual direct access storage device and the data of the virtual direct access storage device are stored together in a single file in a file system; and

wherein the attributes of the virtual direct access storage device include a specification of an internal organization of the virtual direct access storage device for mapping of the data of the virtual direct access storage device from the single file to the data storage, and the specification of the internal organization of the virtual direct access storage device is stored in the single file.

28. The network file server as claimed in claim 27, wherein the specified internal organization of the virtual direct access storage device includes a RAID level.

29. The network file server as claimed in claim 27, wherein the specified internal organization of the virtual direct access storage device includes a striping pattern.

30. The network file server as claimed in claim 27, which includes a snapshot copy facility for copying the data of the virtual direct access storage device over the network concurrent with one of said clients using the block level access protocol over the network to write new data to the virtual direct access storage device.

31. The network file server as claimed in claim 30, wherein the interface is an IP interface, and the network file server is programmed to permit said one of the clients the client to write the new data to the virtual direct access storage device using the block level access protocol over a first TCP/IP connection over the network for the writing of the new data to the virtual direct access storage device, and the network file server is programmed to initiate the copying of the file containing the data of the virtual direct access storage device over the network upon receipt of a command from the client over a second TCP/IP connection over the network.

32. The network file server as claimed in claim 31, wherein the network file server is programmed so that the first TCP/IP connection is concurrent with the second TCP/IP connection.

33. The network file server as claimed in claim 27, wherein the interface is an IP interface, and wherein the network file server includes an IP replication facility for replicating files from the data storage over the network.

34. The network file server as claimed in claim 27, wherein the interface is an IP interface, and the block level access protocol is the Small Computer System Interface (SCSI) protocol.

35. The network file server as claimed in claim 27, wherein the file access protocol is the Network File System (NFS) protocol.

36. The network file server as claimed in claim 27, wherein the file access protocol is the Common Internet File System (CIFS) protocol.

40. A network file server comprising:

data storage;

an interface for coupling the data storage to an IP data network; and

at least one processor programmed for permitting clients in the data network to access the data storage in accordance with a plurality of access protocols;

the data storage containing at least one file for storing file attributes and for storing metadata defining a virtual Small Computer System Interface (SCSI) direct access storage device and for storing data of the virtual direct access storage device;

the access protocols including a block level access protocol for permitting at least one of said clients to access the virtual SCSI direct access storage device over the IP network by accessing the metadata and data of the virtual direct access storage device;

the access protocols including at least one file access protocol for accessing said at least one file; and

the network file server includes a facility for remote replication of said at least one file over the IP network concurrent with write access of said at least one of said clients to the virtual SCSI direct access device over the IP network using the block level access protocol;

wherein the metadata includes attributes of the virtual SCSI direct access storage device, and the attributes of the virtual SCSI direct access storage device and the data of the virtual SCSI direct access storage device are stored together in a single file in a file system; and

wherein the attributes of the virtual SCSI direct access storage device include a specification of an internal organization of the virtual SCSI direct access storage device for mapping of the data of the virtual SCSI direct access storage device from the single file to the data storage, and the specification of the internal organization of the virtual SCSI direct access storage device is stored in the single file.

41. The network file server as claimed in claim 40, wherein the specified internal organization of the virtual SCSI direct access storage device includes a RAID level.

42. The network file server as claimed in claim 40, wherein the specified internal organization of the virtual SCSI direct access storage device includes a striping pattern.

43. The network file server as claimed in claim 40, wherein the network file server is programmed to permit said at least one of said clients to write new data to the virtual SCSI direct access storage device using the block level access protocol over a first TCP/IP connection over the network, and the network file server is programmed to initiate remote replication of said at least one file upon receipt of a command from said at least one of said clients over a second TCP/IP connection over the network.

44. The network file server as claimed in claim 43, wherein the network file server is programmed so that the first TCP/IP connection is concurrent with the second TCP/IP connection.

45. The network file server as claimed in claim 40, wherein said at least one file access protocol includes the Network File System (NFS) protocol.

46. The network file server as claimed in claim 40, wherein said at least one file access protocol includes the Common Internet File System (CIFS) protocol.

47. The network file server as claimed in claim 40, wherein the block-level access protocol includes the Small Computer System Interface (SCSI) protocol.

48. The network file server as claimed in claim 40, wherein the block-level access protocol includes the Small Computer System Interface (SCSI) over IP protocol.

49. The network file server as claimed in claim 40, which includes a snapshot copy facility for creating snapshot copies of said at least one file, and wherein the snapshot copy facility is coupled to the facility for remote replication for transmission of data from the snapshot copies over the IP network concurrent with client write access to the virtual SCSI direct access device over the IP network using the block level access protocol.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.